文章编号:1000-582X(2012)10-136-08

带减函数的连续型软件可靠性验证方案

王学成[1],陆民燕[1],李海峰[1,2],杨日盛[1]

(1. 北京航空航天大学 可靠性与系统工程学院,北京 100091;2. 中航工业综合所 北京 100028)

摘 要:针对连续型安全关键软件可靠性验证测试(SRDT)所需测试时间较长的问题,在现有的基于贝叶斯理论的 SRDT 方案的基础上,提出结合先验信息的基于减函数法的连续型软件可靠性验证测试方案(CBSDF):首先选取连续型软件可靠性参数(如失效率)的典型减函数作为失效率的先验分布密度函数(先验分布);然后根据增长测试阶段后期的失效时间数据(先验信息)计算出先验分布超参数的估计值,进而给出相应的后验分布密度函数,在此基础上得到 CBSDF 的具体形式;最后,将两组真实失效数据集作为先验信息的来源,将 CBSDF 与已有的无先验信息(CBS1)与有先验信息(CBS2)的贝叶斯验证方案进行实例对比研究,计算结果表明:在 SRDT 方案参数相同时,相对于 CBS1 与 CBS2,本论文提出的 CBSDF 可以更为显著地降低所需的验证测试时间,且更适用于高可靠的安全关键连续型软件。

关键词:软件可靠性;可靠性验证;安全关键软件;贝叶斯方法;减函数法;连续执行软件;

中图分类号:TP311.53; V215.7 **文献标志码**:A

Continues software reliability demonstration testing scheme with decreasing function

WANG Xuecheng[1], *LU Minyan*[1], *LI Haifeng*[1,2], *YANG Risheng*[1]

(1. School of Reliability and systems engineering, Beihang University, Beijing 100091, China;
2. Quality Engineering Center, China Aero-Polytechnology Establishment, Beijing 100028, China)

Abstract: The required testing duration of software reliability demonstration testing (SRDT) for the continuous safety-critical software is generally too long to endure. To solve this problem, a new continuous Bayesian-based SRDT scheme (CBSDF) for the safety-critical software is proposed by using the decreasing function method with the prior information. The representative decreasing function of the continuous reliability index (such as the failure rate) is selected to construct the prior distribution density function (PDDF). Then, the estimation values of the hyper-parameters of PDDF are calculated according to the prior information (i. e. the failure data collected in reliability growth testing) to obtain the posterior distribution and the detailed form of CBSDF. By selecting two failure data-sets as the source of prior information, the proposed CBSDF is compared with two existing Bayesian-based schemes (i. e. CBS1 without prior information and CBS2 with prior information). The experimental results show that with the same given reliability index, the proposed CBSDF is more effective than CBS1 and CBS2 by significantly decreasing the required testing time of SRDT and more suitable for the safety-critical software with high reliability.

Key words: software reliability; reliability demonstration; safety-critical software; Bayesian method; decreasing function; continuous software

---

收稿日期:2012-01-10

基金项目:国防科技工业技术基础科研资助项目(Z132010B001)

作者简介:王学成(1985-),女,主要从事软件可靠性验证测试技术方向研究,(Tel)15100575658;(E-mail)xc_w@dse.buaa.edu.cn。

从上世纪 90 年代起，由于软件失效而导致的航空系统异常越发频繁[1]。特别地，安全关键软件（例如机载控制或导航软件）的失效还可能会造成人员生命与财产的灾难性损失。因此，软件尤其是安全关键软件的可靠性已引起众多航空航天公司或研究所的重视[2]。为保证安全关键软件交付时的可靠性水平，在开发之前通常将规定量化的软件可靠性验收指标；因此，在最后的产品验收阶段如何精确、客观、高可信地验证当前软件的可靠性水平是否满足指标要求，是学术界以及工业界一直试图努力解决的问题[3]。软件可靠性验证测试（software reliability demonstration testing，SRDT）就是解决这一问题的重要手段。

SRDT 是用户在接收软件时，确定软件当前的可靠性水平是否满足软件开发合同、开发任务书或需求规格说明中规定的用户的要求而进行的测试，其对于安全关键软件的可靠性保证有着非常重要的意义[4]。验证测试方案是 SRDT 最为重要的组成部分，相关研究也最为丰富[3-13]：Laplace 准则[14]、TRW 法[15]、生命周期测试[16]、MTBF 验证测试[16]、无失效统计测试[17-18]、无先验信息贝叶斯验证测试[7,13]、有先验信息贝叶斯验证测试[19]、概率率序贯测试[16]及单风险序贯验证测试[8]等，以及若干扩展性研究，如考虑失效后果或易测试性的改进方案等[20-25]。SRDT 方案主要有如下 2 种分类方式：根据被测软件的类型，可分为离散型方案（可靠性参数为失效概率或成功率）与连续型方案（可靠性参数为失效率或平均失效间隔时间（MTBF）；根据测试的运行方式，又可分为固定期方案（测试前规定好所需的测试工作量（测试用例数或测试时间）与非固定期方案（所需测试工作量根据测试情况动态调整）。研究内容限定于连续型的固定期 SRDT 方案。

连续型安全关键软件的可靠性参数的指标要求通常较高（例如失效率小于等于 10^{-3} 或 10^{-4}），因此需要大量的验证测试时间。基于贝叶斯理论的 SRDT 方案（简称贝叶斯方案）由于可有效地处理先验信息，能在保证验证结论可信的前提下，显著降低验证测试工作量[3-4]。因此，相关理论研究最为充分且被应用于 NASA 航天类安全关键软件的 SRDT[26]，是目前最为典型且有效的适用于安全关键软件的 SRDT 方案[4]。贝叶斯方案的主要内容是：首先确定待验证可靠性参数的先验分布函数，在此基础上确定后验分布函数，进而给出方案的具体内容。可看出，先验分布函数的确定是贝叶斯方案开展的基础，其对于方案的具体内容与有效性有决定性的影响。目前已有的连续型贝叶斯方案的先验分布构造方法为共轭函数法[27]，即文献[28-29]中选择伽马分布作为失效率的先验分布函数。文献[28]提出一种新的先验分布构造方法——减函数法，该方法的核心思想是选取可靠性参数的减函数作为其先验分布函数的核[27]，符合安全关键软件失效率值较大的可能性小，较小的可能性大的特点[28]，适于构造连续型安全关键软件可靠性参数的先验分布函数。

因此，在已有连续型贝叶斯验证方案[29]的基础上，利用减函数法来构造新的失效率的先验分布函数；从而提出基于减函数法的连续型安全关键软件贝叶斯验证测试方案；最后在 2 组先验信息数据集上将提出的连续型贝叶斯方案与已有的连续型贝叶斯方案进行实例对比研究，以证明新方案的有效性。

1　基于贝叶斯理论的连续型验证测试方案

在已有研究成果[7,19,29]的基础上总结归纳出如下的基于贝叶斯理论的连续型 SRDT 方案形式化构建框架；并在此基础上介绍已有的连续型贝叶斯验证测试方案。

1.1　连续型贝叶斯验证测试方案的形式化框架

设测试过程中累积探测失效数 r 的条件分布密度函数为 $f(r|\lambda)$，其中 λ 表示待验证的软件可靠性参数（如失效率或 MTBF 等），为便于讨论，限定 λ 表示软件失效率；假设 λ 的先验分布密度函数为 $h(\lambda)$，则 (r,λ) 的联合分布密度函数为

$$g(r,\lambda) = f(r \mid \lambda) \cdot h(\lambda)。 \tag{1}$$

由式(1)可得失效数 x 的无条件密度函数为

$$p(r) = \int g(r,\lambda)\mathrm{d}\lambda = \int f(r \mid \lambda) \cdot h(\lambda)\mathrm{d}\lambda。 \tag{2}$$

进而得到 λ 的后验密度函数为

$$\omega(\lambda \mid r) = \frac{g(r,\lambda)}{p(r)} = \frac{f(r \mid \lambda) \cdot h(\lambda)}{\int f(r \mid \lambda) \cdot h(\lambda)\mathrm{d}\lambda}。 \tag{3}$$

若给定的 SRDT 方案参数为 (λ_0, c, r)，λ_0 表示给定的失效率指标，c 表示置信水平，r 表示测试过程中可容忍的失效数。则所需验证测试时间即为满足下式的 w 的最小值

$$P(\lambda \leqslant \lambda_0) = \int_0^{\lambda_0} \omega(\lambda \mid r, w)\mathrm{d}\lambda \geqslant c。 \tag{4}$$

1.2　已有的连续型贝叶斯方案

1）无先验信息的连续型贝叶斯方案（CBS1）

该方案选取 Gamma(1,0) 作为失效率 λ 的先验分布密度函数，则若在时间 $(0,t]$ 内发现 r 个失效，λ

的后验分布密度函数为

$$\omega(\lambda \mid r,t,1,0) = \text{Gamma}(1+r,t)。 \quad (5)$$

给定方案参数为(λ_0,c,r),结合公式(4)和(5)即可得到CBS1所需的验证测试时间T。

2)有先验信息的连续型贝叶斯方案(CBS2)

该方案选取Gamma(a,b)为λ的先验分布密度函数,再利用先验信息获得超参数(a,b)的估计值$(\hat{a},\hat{b})$,则若在时间$(0,t]$内发现r个失效,λ的后验分布密度函数为

$$\omega(\lambda \mid r,t,\hat{a},\hat{b}) = \text{Gamma}(\hat{a}+r,\hat{b}+t)。 \quad (6)$$

给定方案参数为(λ_0,c,r),结合公式(4)和(6)即可得到CBS1所需的验证测试时间T。

2 基于减函数法的连续型贝叶斯方案

首先确定基于减函数法的失效率的先验分布密度函数及相应的后验分布密度函数,在此基础上分别提出无先验信息时的基于减函数法的连续型贝叶斯方案(NCBSDF)及有先验信息时的基于减函数法的连续型贝叶斯方案(CBSDF)。

2.1 先验分布与后验分布

首先引入分布密度的核的概念[27]:

定义:设$f(x)$是随机变量X的分布密度函数,若$f(x)=Ag(x)$,A是与x无关的常数,$g(x)$是与x有关的部分,则称$g(x)$为$f(x)$的核,记为

$$f(x) \propto g(x)。 \quad (7)$$

假设连续型软件的失效率λ是一个随机变量,依据减函数法,选取λ的一个典型减函数$e^{-a\lambda}$作为其先验分布密度函数的核,即有

$$h(\lambda) \propto \mathrm{e}^{-a\lambda}, \quad (8)$$

其中a是待估计的分布超参数。

由式(8)可得λ的先验分布密度函数为

$$h(\lambda) = A\mathrm{e}^{-a\lambda}。 \quad (9)$$

根据分布密度函数的性质可知

$$\int_0^{\infty} h(\lambda)\mathrm{d}\lambda = 1。 \quad (10)$$

将式(9)代入式(10),化简后可知$A=a$。因此,重写式(9)可得λ的先验分布密度函数为

$$h(\lambda) = a\mathrm{e}^{-a\lambda}。 \quad (11)$$

假设连续型软件在时间间隔$(0,t]$内的失效次数X等于r的概率,是失效率λ的条件概率,且服从参数为λt的泊松分布[7,29],则有

$$g(X=r \mid \lambda) = \frac{(\lambda t)^r}{r!}\mathrm{e}^{-\lambda t}。 \quad (12)$$

结合式(11)与(12),即可获得失效次数X及失效率λ的联合分布如下

$$g(X=r,\lambda) = a\frac{(\lambda t)^r}{r!}\mathrm{e}^{-\lambda(a+t)}。 \quad (13)$$

由式(13),可获得失效次数X的边缘分布为

$$g(X=r) = \int_0^{+\infty} g(X=r,\lambda)\mathrm{d}\lambda = \int_0^{+\infty} a\frac{(\lambda t)^r}{r!}\mathrm{e}^{-\lambda(a+t)}\mathrm{d}\lambda = \frac{at^r}{r!}\cdot\frac{\Gamma(r+1)}{(a+t)^{r+1}}, \quad (14)$$

其中,$\Gamma(x)$是Gamma函数[30],其形式如下

$$\Gamma(x) = \int_0^{\infty} t^{x-1}e^{-t}\mathrm{d}t。 \quad (15)$$

若x是正整数,则$\Gamma(x)=(x-1)!$,由此可进一步化简式(14)为

$$g(X=r) = \frac{at^r}{(a+t)^{r+1}}。 \quad (16)$$

假设软件连续运行时间t之后,共发现r个失效,则可获得失效率λ的后验分布密度函数为

$$\omega(\lambda \mid r,t,a) = \frac{g(X=r,\lambda)}{g(X=r)} = \frac{(a+t)^{r+1}}{r!}\lambda^r\mathrm{e}^{-\lambda(a+t)}。 \quad (17)$$

2.2 无先验信息(NCBSDF)

当没有任何无先验信息时,只能选取均匀分布作为失效率λ的先验密度分布函数,也即有$h(\lambda)=a\mathrm{e}^{-a\lambda}=1$。将该分布函数代入式(13),再经过与式(14)、(16)相同的推导,即可获得λ的后验分布密度函数为

$$\omega(\lambda \mid r,t) = \text{Gamma}(1+r,t)。 \quad (18)$$

则对于给定的SRDT方案参数(λ_0,c,r),满足下式的最小t值即为对应的验证测试时间T

$$P(\lambda \leqslant \lambda_0) = \int_0^{\lambda_0} \text{Gamma}(r+1,t)\mathrm{d}\lambda \geqslant c。 \quad (19)$$

由式(18)、(19)可知,当无先验信息时,基于减函数法的贝叶斯验证方案与传统无先验信息贝叶斯方案(CBS1)是完全相同的。

2.3 有先验信息(CBSDF)

当具有先验信息时,失效率的先验分布密度函数式(11)中的超参数a将不再是一个定值,而是一个由先验信息数据来确定的待评估变量。通常在安全关键软件验收之前,都会经历较长时间的软件可靠性增长测试,期间收集到的若干失效间隔时间数据(记为$T_1,\cdots,T_n$)是目前最为可信且可用的先验信息数据来源[7,29],所以应选取增长测试阶段后期收集到的若干失效时间间隔数据作为先验信息数据,以保证尽量准确地确定失效率λ的先验分布密度函数。根据这种失效信息数据来确定超参数a的具体方法如下所示。

由式(16)可获得软件在时间间隔$(0,t]$内的失

效次数 X 为 r 的期望如下

$$E(x)=\sum_{r=0}^{+\infty}r\cdot g(X=r)=\sum_{r=0}^{+\infty}\frac{art^r}{(a+t)^{r+1}}=\frac{t}{a}。\tag{20}$$

由式(20)可知,超参数 a 的估计值可由时间 t 以及时间间隔(0,t]内的失效数期望值 $E(x)$来确定。所以需要先将失效间隔时间样本序列(T_1,T_2,…,T_n)转化为失效数形式的样本序列,再代入式(20),即可获得超参数 a 的估计值。

首先假设时刻 t_φ 为一个较大的时间数值(相对于失效间隔序列 T_1,T_2,…,T_n 来说,是一个较大的时间点),则 T_i 在时间(0,t_φ]内对应的失效数样本值 s_i 为 t_φ/T_i,从而将 T_1,T_2,…,T_n 转化为如下的失效数样本序列

$$\{s_i\}_{i=1}^n=(t_\varphi/T_i)_{i=1}^n。\tag{21}$$

根据式(20)与(21)可得如下方程

$$E(x)=\frac{1}{n}\sum_{i=1}^{n}s_i=t/\hat{a}。\tag{22}$$

对上式进行求解,即可得超参数 a 的估计值为

$$\hat{a}=\frac{t_\varphi}{\frac{1}{n}\sum_{i=1}^{n}s_i}。\tag{23}$$

将式(21)代入式(23),进行化简后可得

$$\hat{a}=\frac{n}{\sum_{i=1}^{n}\frac{1}{T_i}}。\tag{24}$$

由式(24)可知,时刻 t_φ 的取值对超参数 a 的估计结果并无影响,该结论也同样适用于 CBS2。

若根据先验信息数据,由式(24)获得超参数 a 的估计值$\hat{a}$,则由式(16)可知,失效率 λ 的后验分布密度函数为

$$\omega(\lambda\mid r,t,\hat{a})=\frac{(\hat{a}+t)^{r+1}}{r!}\lambda^r e^{-\lambda(\hat{a}+t)}。\tag{25}$$

对于给定的 SRDT 方案参数(λ_0,c,r),满足下式的最小 t 值即为对应的所需验证测试时间 T

$$P(\lambda\leqslant\lambda_0)=\int_0^{\lambda_0}\omega(\lambda\mid r,t,\hat{a})\mathrm{d}\lambda\geqslant c。\tag{26}$$

2.4　容许失效数 r 不为 0 时的先验动态整合方法

先验动态整合方法是指:若在规定的验证测试时间内发生的失效数 R 大于允许的失效数 r,即软件没有通过验证测试,则此时可结合当前的测试结果,即累积失效数 R,来动态确定后续测试所需的验证测试时间,而不需要在导致失效的缺陷改正以后重新开始 SRDT,从而达到降低整体测试时间的目的。

文献[7, 29]中针对容许失效数 $r=0$ 时的先验动态整合方法进行了讨论。在此基础上,将给出容许失效数 r 不为 0 时的先验动态整合方法。假设容许的失效数为 $r(r\neq0)$,在有限测试资源下允许的最长测试时间为 T,则容许失效数 r 不为 0 时的动态整合步骤如下

步骤 1:根据给定的 SRDT 方案参数(λ_0,c,r),及验证测试方案,如式计算出所需的验证测试时间为 T_r;

步骤 2:执行软件可靠性验证测试至时刻 T_r,若此过程中发现的缺陷数 r' 小于等于 r,则接收该软件,转步骤 5;否则转步骤 3;

步骤 3:将缺陷数 r' 代入式(26)得到下式,则允许 r' 个失效的验证测试时间为 T_r' 为满足下式的最小 t 值

$$P(\lambda\leqslant\lambda_0)=\int_0^{\lambda_0}\omega(\lambda\mid r',t,\hat{a})\mathrm{d}\lambda\geqslant c,\tag{27}$$

若 $T_r'>T$ 则拒收该软件,转步骤 5,否则转步 4;

步骤 4:继续进行时长为 $T_r'-T_r$ 的验证测试,若此过程中没有发生失效,则接收该软件,转步骤 5;否则,若此过程中发生的失效数为 k,则令 $T_r=T_r'$,$r'=r'+k$,重新执行步骤 3;

步骤 5:结束验证测试。

3　实例验证

该实例选择某安全关键软件的 2 组真实失效数据集作为先验信息数据的来源,同时依据 2 组失效数据集的特点来确定 SRDT 方案参数,进而将提出的基于减函数法的有先验信息的贝叶斯验证方案(CBSDF)与 2 种经典的贝叶斯验证方案(CBS1 与 CBS2)进行实例对比研究,以验证新方案 CBSDF 的有效性与适用性,即在 SRDT 方案参数相同的前提下,CBSDF 所需要的验证测试时间要显著短于 CBS1 和 CBS2。

3.1　实例介绍

1)先验信息数据的来源

本实例选取 2 组真实失效数据集"SYS1"与"SYS2"[30]中最后阶段的 10 个失效间隔数据 T_1,T_2,…,T_{10}(如表 1 所示)作为先验信息数据的来源。这两组失效数据集均来自于美国 Rome 航空发展中心的某实时控制系统软件的可靠性增长测试过程。该实时控制系统软件属于连续型安全关键类软件,可作为本文研究成果的应用对象。假设时刻 t_φ 为 100 000 h(SYS1 与 SYS2 中的最大失效间隔时间均不超过 10 000 h,因此 t_φ 相对于 T_1,T_2,…,T_{10} 是一个较大的数值),则 2 组失效数据集中的 T_1,T_2,…,

T_{10}对应于时间(0, t_φ]内的经验失效数序列$\{s_i\}_{i=1}^{10}=(t_\varphi/T_i)_{i=1}^{10}$如表 1 所示;

2)SRDT 方案参数(λ_0,c,r)

考虑到安全关键软件的特点,本实例中将失效率λ_0定为10^{-3},置信度水平c定为 0.99,允许的失效数r分别设置为 0,1,2,3,4,5;

表 1 先验信息数据(时间单位:h)

SYS1		SYS2	
T_i	s_i	T_i	s_i
1 071	93	2 175	45
371	269	1 866	53
790	126	2 716	36
6 150	16	1 520	65
3 321	30	725	137
1 045	95	490	204
648	154	1 194	83
5 485	18	994	100
1 160	86	3 281	30
1 864	53	3 902	25

3.2 基于 SYS1 数据集的计算结果与分析

1)方案 CBS1 与 CBS2 的计算结果

首先,根据表 1 中列举的 SYS1 先验信息数据,结合文献[29]中介绍的方案 CBS2 先验分布超参数估计方法,可得到 CBS2 先验分布的 2 个超参数(式 6 中的 a 和 b)估计值分别为$\hat{a}=1.7$,$\bar{b}=1\ 807$,则 CBS2 的先验分布密度函数为

$$h(\lambda)=\frac{1\ 807^{1.7}\lambda^{0.7}e^{-1\ 807\lambda}}{\Gamma(1.7)}。\tag{28}$$

相应地,CBS2 的后验分布密度函数为

$$\omega(\lambda\mid r,t,\hat{a},\bar{b})=\mathrm{Gamma}(1.7+r,1807+t)。\tag{29}$$

将式(5)与式(29)分别代入式(4),再根据确定的方案参数(λ_0,c,r),即可分别获得 CBS1 与 CBS2 方案在不同允许失效数r($r=0,1,2,3,4,5$)时的验证测试时间(见表 2);

2)CBSDF 的计算结果

根据表 1 中列举的 SYS1 先验信息数据,结合式(23)中介绍的超参数估计方法,可获得方案 CBSDF 先验分布密度函数中的超参数的估计值为$\hat{a}=1\ 064$,则先验分布密度函数为

$$h(\lambda)=1\ 064e^{-1\ 064\lambda}。\tag{30}$$

相应地,CBSDF 的后验分布密度函数为

$$\omega(\lambda\mid r,t,\hat{a})=\frac{(1\ 064+t)^{r+1}}{r!}\lambda^r e^{-\lambda(1\ 064+t)}。\tag{31}$$

将式(31)代入式(4),再根据方案参数(λ_0,c,r),即可分别获得 CBSDF 方案在不同允许失效数r($r=0,1,2,3,4,5$)时的验证测试时间(见表 2)

表 2 各方案所需的验证测试时间(SYS1)

r	CBS1	CBS2	CBSDF
0	4 605.2	4 258.4	3 541.3
1	6 638.4	6 083.3	5 574.5
2	8 405.9	7 754.1	7 342.1
3	10 045.1	9 334.4	8 981.3
4	11 604.6	10 853.2	10 540.8
5	13 108.5	12 326.8	12 044.6

3)计算结果分析

根据表 2 的计算结果,可知

a)对于不同的允许失效数r($r=0,1,2,3,4,5$),有先验信息的贝叶斯方案(即 CBS2 与 CBSDF)所需的验证测试时间均显著小于无先验信息的贝叶斯方案(即 CBS1),例如当$r=0$时,CBS1 所需的验证测试时间为 4 605.2 h,而 CBS2 与 CBSDF 所需的验证测试时间则分别为 4 258.4 h 与 3 541.3 h。这表明,若能够获得有效的先验信息数据,则有先验信息的贝叶斯方案可以更加准确的描述失效率的分布情况,进而显著降低所需验证测试时间。

b)对于不同的允许失效数r($r=0,1,2,3,4,5$),CBSDF 所需的验证测试时间均显著小于 CBS1 与 CBS2,例如,当$r=0$时,CBS1 与 CBS2 所需的验证测试时间分别为 4 605.2 h 与 4 258.4 h,而 CBSDF 所需的验证测试时间则为 3 541.3 h。即相对于方案 CBS1 与 CBS2,CBSDF 所降低的测试时间分别为 1 063.9 h、717 h,也即所降低的程度分别为 23%与 17%。这表明在具有相同先验信息时,提出的基于减函数法的贝叶斯方案可准确地描述失效率的先验分布情况,进而可显著降低所需验证测试时间。

c)值得注意的是,在表 2 中,随着容忍失效数r的增加,提出的 CBSDF 方案所需的验证测试时间与传统的 CBS1 和 CBS2 方案所需的验证测试时间之间的差距变得逐渐变小,也即 CBSDF 方案在降

低所需验证测试时间上的显著性随着容忍失效数 r 的增加而逐渐降低。例如相对于 CBS2,当 $r=5$ 时,CBSDF 降低所需测试时间的程度由 $r=0$ 时的 17%降低至 2.3%。针对此现象,有如下推论:CBSDF 方案在安全关键软件的可靠性要求较高时,效果更加明显。也即 CBSDF 方案可能更加适用于高可靠的安全关键软件。因为一般来说,高可靠的安全关键软件在验证测试过程允许的失效数是非常少的,甚至不允许发生失效。该推论产生的可能原因是:CBSDF 方案是基于减函数的思想提出的,即"选取可靠性参数的减函数作为其先验分布函数的核,符合安全关键软件失效率值较大的可能性小,较小的可能性大的特点",因此当可靠性要求越高时,安全关键软件失效率值较大的可能性就越小,较小的可能性就越大,十分符合减函数的核心思想,故利用减函数所构造的失效率先验分布函数就越加准确,进而更为显著地降低所需的验证测试时间。

3.3　基于 SYS2 数据集的计算结果与分析

1)方案 CBS1 与 CBS2 的计算结果

根据表 1 中列举的 SYS2 先验信息数据,可得到 CBS2 先验分布的 2 个超参数的估计值分别为 $\hat{a}=2.2$,$\overline{b}=2\ 793$,则此时,CBS2 的先验分布密度函数为

$$h(\lambda)=\frac{2\ 793^{2.2}\lambda^{1.2}\mathrm{e}^{-2\ 793\lambda}}{\Gamma(2.2)}。\tag{32}$$

分别获得 CBS1 与 CBS2 在不同允许失效数 r 时的验证测试时间(见表 3):

2)CBSDF 的计算结果

根据表 1 中列举的 SYS2 先验信息数据,可获得方案 CBSDF 先验分布密度函数中的超参数的估计值为 $\hat{a}=1\ 285$,则先验分布密度函数为

$$h(\lambda)=1\ 285\mathrm{e}^{-1\ 285\lambda}。\tag{33}$$

可分别获得 CBSDF 在不同允许失效数 r 时的验证测试时间(见表 3)

表 3　各方案所需的验证测试时间(SYS2)

r	CBS1	CBS2	CBSDF
0	4 605.2	4 164.3	3 319.8
1	6 638.4	5 903.9	5 353
2	8 405.9	7 526.9	7 120
3	10 045.1	9 075	8 759
4	11 604.6	10 571.1	10 319.3
5	13 108.5	12 027.1	11 823.1

3)计算结果分析

根据表 3 的计算结果,可知

a)表 2 与表 3 中的 CBS1 计算结果完全一样,即 CBS1 只与方案参数有关,而不考虑任何先验信息。所以 CBS1 是一种最为保守的验证测试方案,若无法获得任何先验信息或者先验信息的来源不可靠时,可以考虑采用 CBS1。

b)对于不同的允许失效数 $r(r=0,1,2,3,4,5)$,CBSDF 所需的验证测试时间依然均显著小于 CBS1 与 CBS2,例如,当 $r=0$ 时,CBS1 与 CBS2 所需的验证测试时间分别为 4 605.2 h 与 4 164.3 h,而 CBSDF 所需的验证测试时间则为 3 319.8 h。即相对于方案 CBS1 与 CBS2,CBSDF 所降低的测试时间分别为 1 285.4 h(28%)、844.5 h(20%)。从而再次证明了 CBSDF 的有效性。

c)值得注意的是,在表 3 中,随着容忍失效数 r 的增加,提出的 CBSDF 方案所需的验证测试时间与传统的 CBS1 和 CBS2 方案所需的验证测试时间之间的差距变得逐渐变小。这也再次证明了提出的推论,即 CBSDF 方案可能更加适用于高可靠的安全关键软件。

3.4　进一步的分析

为进一步证明推论:CBSDF 可能更适用于安全关键类软件,将针对 CBS2 与 CBSDF 的计算结果随失效率指标 λ_0 在 $[10^{-4},10^{-3}]$ 内的变化情况进行讨论。

1)方案参数:$c=0.99$,允许失效数 $r=0$,失效率指标 λ_0 的取值空间为 $[10^{-4},10^{-3}]$。

2)先验信息数据为 SYS1 与 SYS2,当 λ_0 由 10^{-4} 逐步递增为 10^{-3} 时,分别计算在不同 λ_0 下,CBS2 与 CBSDF 所需的验证测试时间 T(如图 1 和图 2 所示),其中横坐标表示失效率 λ_0,纵坐标表示所需的验证测试时间 T。

3)由图 1 与图 2 可以看出:a)随着 λ_0 值在 $[10^{-4},10^{-3}]$ 区间内逐渐增大,CBS2 及 CBSDF 所需的验证测试时间 T 均呈现一种递减趋势,符合失效率指标越高所需的验证测试时间越短这一显而易见的结论;b)在 $\lambda_0\in[10^{-4},10^{-3}]$ 时,CBSDF 所需的验证测试时间始终小于 CBS2,证明 CBSDF 比 CBS2 具有更为显著的有效性与适用性;c)失效率指标较小时($10^{-4}<\lambda_0<4\times10^{-4}$),CBSDF 所需的验证测试时间要显著小于 CBS2,但随着 λ_0 的逐渐增大,CBSDF 的曲线走势较缓,与 CBS2 的差距越来越小。从而证明相对于 CBS2,CBSDF 在高可靠安全关键软件上的效果更显著。

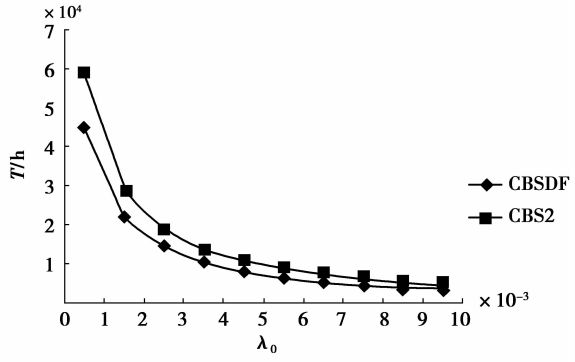

图 1 不同 λ_0 下的验证测试时间(SYS1)

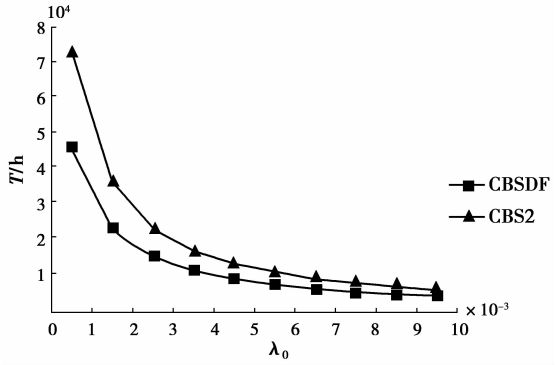

图 2 不同 λ_0 下的验证测试时间(SYS2)

4 结 论

提出一种基于减函数法的连续型安全关键软件的可靠性贝叶斯验证测试方案(CBSDF)。实例验证结果表明:1)CBSDF 充分考虑了连续型安全关键软件失效率大的可能性小,而小的可能性大这一特点,因此,利用减函数法来指导失效率先验分布的构造是合理的;2)方案参数相同时,与两种贝叶斯方案(CBS1 与 CBS2)相比,CBSDF 的有效性与适用性更为优秀,即可以更为显著的降低所需的验证测试时间。这也证明了,若能够更为合理与准确的使用先验信息,可进一步降低验证测试时间。

参考文献:

[1] Hecht M, Buettner D. Software testing in space programs[J]. Crosslink: the aerospace Corporation magazine of advances in Aerospace Technology, 2005, 6(3):31-35.

[2] Li S M, Yin Q, Guo P, et al. A hierarchical mixture model for software reliability prediction[J]. Applied Mathematics and Computation, 2007, 185(2): 1120-1130.

[3] 覃志东,雷航,桑楠,等. 安全关键软件可靠性验证测试方法研究[J]. 航空学报, 2005, 26(3):334-339.
QIN Zhidong, LEI Hang, SANG Lan, et al. Study on the reliability demonstration testing method for safety-critical software[J]. Acta Aeronautica Et Astronatica Sinica, 2005, 26(3): 334-339

[4] Tal O, Bendell A, Mccollin C. A comparison of methods for calculating the duration of software reliability demonstration testing, particularly for safety-critical systems[J]. Quality and Reliabiliy Engineering International, 2000, 16(1): 59-62.

[5] Tal O, Mccollin C, Bendell A. An optimal statistical testing policy for software reliability demonstration of safety-critical systems [J]. European Journal of Operational Research, 2002, 137(3): 544-557.

[6] Howden W E. Good enough versus high assurance software testing and analysis methods[C]// Proceedings of the Third IEEE International High-Assurance Systems Engineering Symposium, November 13-14, 1998, Washington, DC, USA. Washington, DC: IEEE Computer Society, 1998: 166-175.

[7] Littlewood B, Wright D. Some conservative stopping rules for the operational testing of safety-critical software [J]. IEEE Transactions on Software Engineering, 1997, 23(11): 673-683.

[8] Tal O, Mccollin C, Bendell T. Reliability demonstration for safety-critical systems [J]. IEEE Transactions on Reliability, 2001, 50(2): 194-204.

[9] Sawada K, Sandoh H. Continuous model for software reliability demonstration testing considering damage size of software failures[J]. Mathematical and Computer Modelling: An International Journal, 2000, 31(10/11/12):321-326.

[10] 杨仕平, 桑楠, 熊光泽. 安全关键软件的防危性测评技术研究[J]. 计算机学报, 2004, 27(4): 442-450.
YANG Shiping, SANG Nan, XIONG Guangze. Research on safety testing and evaluation technology of safety critical software [J]. Chinese Journal of Computers, 2004, 27(4): 442-450.

[11] 覃志东,雷航,桑楠,等. 连续执行软件可靠性验证测试方法[J]. 计算机科学, 2005,32(6):202-205.
QIN Zhidong, LEI Hang, SANG Nan, et al. Reliability demonstration testing method for continuous execution software [J]. Computer Science, 2005, 32(6):202-205.

[12] Miller W M, Morell L J, Noonan R E, et al. Estimating the probability of failure when testing

reveals no failures[J]. IEEE Transactions on Software Engineering, 1992, 18(1): 33-43.

[13] Littlewood B, Strigini L. Assessment of ultra-high dependability for software-based systems [J]. Communications of ACM, 1993, 36 (11): 69-80

[14] Feller W. An introduction to probability theory and its applications [M]. 3rd ed. New York: Wiley,,1968.

[15] Thayer T A, Lipow M, Nelson E C. Software reliability [M]. Amsterdam, Holanda: North-Holland, 1978.

[16] MIL-HDBK-781A, Military Handbook: reliability test methods, plans and environments for engineering, development qualification, and production[R]. DoD: Washington D. C., 1996.

[17] Parnas D L, Schouwen A J V, Kwan S P. Evaluation of safety-critical software[J]. Communications of the ACM, 1990, 33(6): 636-648.

[18] Miller W K, Morell L J, Noonan R E, et al. Estimating the probability of failure when testing reveals no failures[J]. IEEE Transactions on Software Engineering, 1992, 18(1): 33-43.

[19] 覃志东. 高可信软件可靠性和防危性测试与评价理论研究[D]. 成都:电子科技大学博士学位论文,2005.

[20] 李秋英, 姜梦岑. 软件可靠性验证测试最小测试量的必要条件[J]. 北京航空航天大学学报, 2010, 36(2): 239-244.
LI Qiuying, JIANG Mengcen. Analysis of necessary condition for minimal software reliability demonstration test suite [J]. Journal of Beijing University of Aeronautics and Astronautics, 2010, 36(2): 239-244.

[21] Kuball S, May J. Test-adequacy and statistical testing: combining different properties of a test-set [C] // Proceedings of the 15th IEEE International Symposim on Software Reliability Engineering, November 2-5, 2004, California, USA. Piscataway: IEEE Press, 2004: 161-172.

[22] Ammann P E, Brilliant S S, Knight J C. The effect of imperfect error detection on reliability assessment via life testing [J]. IEEE Transactions on Software Engineering, 1994, 20(2): 142-148.

[23] Bertolino A, Strigini L. On the use of testability measures for dependability assessment [J]. IEEE Transactions on Software Engineering, 1996, 22(2): 97-108.

[24] 赵亮, 王建民, 孙家广. 软件易测性和软件可靠性关系研究 [J]. 计算机学报, 2007, 30(6): 986-992.
ZHAO Liang, WANG Jianmin, SUN Jiaguang. Study on the relationship between software testability and reliability[J]. Chinese Journal of Computers, 2007, 30(6): 986-992.

[25] 闵庆欢. 软件可靠性验证测试中降低测试用例量方法研究[D]. 南京:南京理工大学硕士学位论文, 2009.

[26] Bojan Cukic, Diwakar Chakravarthy. Bayesian framework for reliability assurance of a deployed safety critical system [C]. The 5th IEEE International Symposim on High Assurance Systems Engineering [A]. 2000: 321-329

[27] 张尧庭,陈汉峰. 贝叶斯统计与推断[M]. 北京:科学出版社, 1991.

[28] 韩明. 基于无失效数据的可靠性参数估计[M]. 北京:中国统计出版社, 2005.

[29] Wikipedia. Gamma function [EB/OL]. [4]http://en.wikipedia.org/wiki/Gamma_function

[30] Lyu M R. Handbook of Software Reliability Engineering[M]. Washington, DC: IEEE Computer Society Press, 1996.

（编辑　侯　湘）